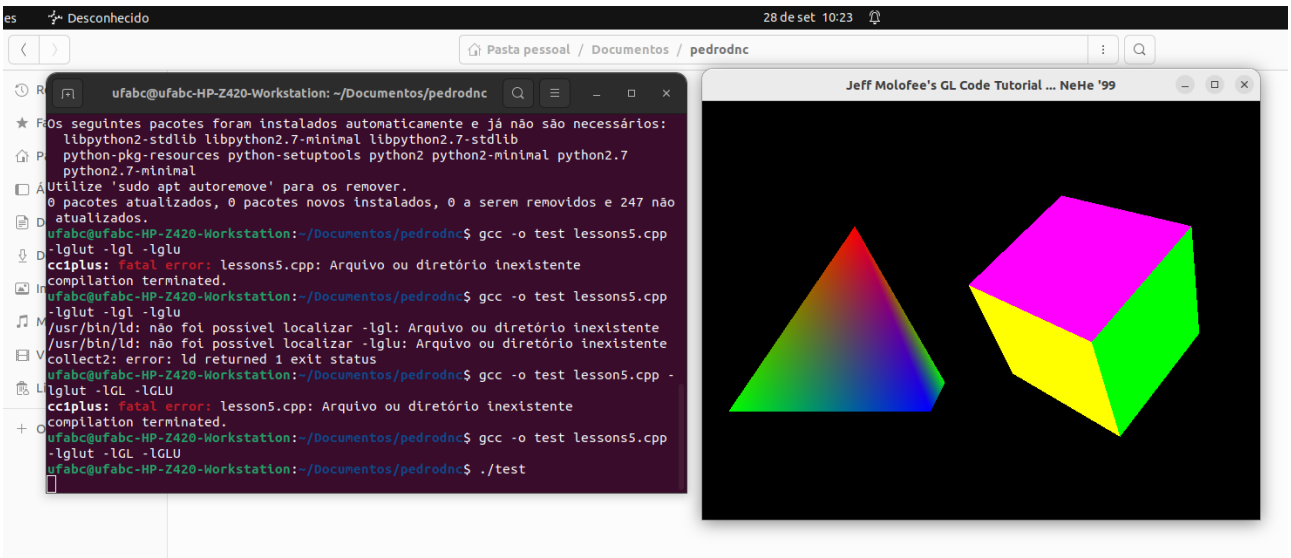
Universidade Federal do ABC
Fundamentos de Processamento Gráfico 2023.3
Prof. Celso Setsuo Kurashima
Andressa Guimaraes Benedicto 11201810280
Aryanne Gramacho Acosta 11201722556
Heitor Rodrigues Savegnago 11077415
Pedro Domingos Napole Certo 11201722682

Laboratório 01 - OpenGL Linux

O OpenGL é uma API gráfica que permite a programação das placas gráficas, ou GPUs. É uma ferramenta essencial para construção de visualizações tridimensionais em computadores.

Através de um SO baseado em Linux (Ubuntu, por exemplo), pode-se seguir os passos aqui descritos para instalação das ferramentas e configuração do ambiente para trabalhar com a API gráfica em questão.

1. Criação de um diretório:
    a. No terminal, digitar o comando:
    $ mkdir lab01_OpenGL
2. Para instalação das ferramentas, deve-se utilizar o comando de "super-usuário": **sudo** para atualização e instalação (rodar uma linha por vez):
    a. $ sudo apt-get update
    b. $ sudo apt-get install build-essential
    c. $ sudo apt upgrade
    d. $ sudo apt-get install g++ freeglut3-dev buildessential libx11-dev libxmu-dev libxi-dev libglu1-mesa libglu1-mesa-dev
    e. $ sudo apt-get install geany
3. Utilizar o arquivo de teste lessons5.cpp para compilação inicial (verificação da instalação dos pacotes e ambiente).
    a. Compilar como um programa C++ comum, acrescentado dos comandos:
        i. -lglut
        ii. -lGL
        iii. -lGLU
        gcc -o first_gl lessons5.cpp -lglut -lGL -lGLU
4. executar e testar:
    $ ./first_gl
5. resultado esperado:



Complementar ao OpenGL, existe outra ferramenta de cunho gráfico que também é amplamente utilizada para aplicações com visualizações gráficas, o OpenCV. Diferentemente do primeiro, este é mais utilizado para aplicações de visão computacional (detecção de objetos, por exemplo) e possui maior comunidade/suporte para a linguagem Python.

Utilizando o mesmo sistema operacional, pode-se instalar a ferramenta através dos comandos **sudo**:

1. Instalação:
    a. $ sudo apt install build-essential cmake git pkg-config libgtk-3-dev \
    libavcodec-dev libavformat-dev libswscale-dev libv4l-dev \
    libxvidcore-dev libx264-dev libjpeg-dev libpng-dev libtiff-dev \
    gfortran openexr libatlas-base-dev python3-dev python3-numpy \
    libtbb2 libtbb-dev libdc1394-dev libopenexr-dev \
    libgstreamer-plugins-base1.0-dev libgstreamer1.0-dev
2. Clonar projeto base para teste inicial (GitHub) em alguma pasta:
    a. $ mkdir opencv_intro
    b. $ cd opencv_intro
    c. $ git clone https://github.com/opencv/opencv.git
    d. $ git clone https://github.com/opencv/opencv_contrib.git
3. Criar diretório dentro da pasta raiz do projeto clonado, para build da aplicação:
    a. $ mkdir opéncv_build
    b. $ cd opencv_build
    c. $ cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_C_EXAMPLES=ON \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D OPENCV_GENERATE_PKGCONFIG=ON \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_build/opencv_contrib/modules \
    -D BUILD_EXAMPLES=ON ..
4. Iniciar compilação
    a. $ nproc
    b. $ make -j12
    c. $ sudo make install
5. Verificar instalação da biblioteca no Python (versão 3):
    a. $ python3 -c “import cv2; print(cv2.__version__)
    b. $pkg-config –modversion opencv4
6. Por fim, caso as etapas tenham sido concluídas com sucesso, rodar o programa:
    a. $ python3 teste_messi.py
7. Resultado final esperado: